**PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**ACTIVIDAD BESE DE DATOS**

**ANALISIS Y DESARROLLO DE SOFTWARE 228118**

**BRAYAN TORO**

**FICHA 2672883**

**MONICA PENAGOS MARTINEZ**

**BASE DE DATOS**

**BOGOTA 2023**

**MODULO VENTAS**

1.Este trigger se activará automáticamente después de cada inserción en la tabla "ventas" y capturará la información

requerida de la venta. Luego, insertará una notificación ficticia en la tabla "notificaciones".

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

DELIMITER //

CREATE TRIGGER VentaInsertTrigger

AFTER INSERT ON ventas FOR EACH ROW

BEGIN

DECLARE tipo\_servicio VARCHAR(255);

DECLARE usuario\_id INT;

DECLARE cantidad INT;

DECLARE precio\_venta DECIMAL(10, 2);

DECLARE fecha\_venta DATE;

SELECT NEW.tipo\_servicio, NEW.usuario, NEW.cantidad, NEW.precio\_venta, NEW.fecha\_venta

INTO tipo\_servicio, usuario\_id, cantidad, precio\_venta, fecha\_venta;

INSERT INTO notificaciones (mensaje)

VALUES (CONCAT('Nueva venta: Tipo de Servicio - ', tipo\_servicio, ', Usuario - ', usuario\_id,

', Cantidad - ', cantidad, ', Precio de Venta - ', precio\_venta, ', Fecha de Venta - ', fecha\_venta));

END;

//

DELIMITER ;

SELECT \* FROM notificaciones;

2.El procedimiento almacenado "CalcularTotalVentasDeUsuario" toma como entrada el ID de un usuario y utiliza

una consulta SQL para calcular la suma total de las ventas realizadas por ese usuario.

La consulta selecciona todas las ventas del usuario especificado y suma los precios de venta.

El resultado se almacena en la variable "total" utilizando la cláusula "INTO" obtener la suma total de sus ventas.

Una captura de pantalla de una computadora

Descripción generada automáticamente

DELIMITER //

CREATE PROCEDURE CalcularTotalVentasDeUsuario(IN usuario\_id INT, OUT total DECIMAL(10, 2))

BEGIN

SELECT SUM(venta.precio\_venta)

INTO total

FROM ventas AS venta

WHERE venta.usuario = usuario\_id;

END //

DELIMITER ;

-- Ejecutar el procedimiento y obtener el total de ventas para el usuario con ID 1

CALL CalcularTotalVentasDeUsuario(1, @total);

-- Obtener el resultado

SELECT @total;

3. Este procedimiento almacenado se ejecutará en la tabla "productos" y mostrará el "id\_producto", "nombre", "precio\_compra", "cantidad" y la categoría calculada según el valor total (precio\_compra \* cantidad) del producto.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

DELIMITER //

CREATE PROCEDURE ClasificarProductos()

BEGIN

SELECT

id\_producto,

nombre,

precio\_compra,

cantidad,

CASE

WHEN precio\_compra \* cantidad < 50 THEN 'Producto de bajo valor'

WHEN precio\_compra \* cantidad >= 50 AND precio\_compra \* cantidad < 100 THEN 'Producto de valor medio'

WHEN precio\_compra \* cantidad >= 100 THEN 'Producto de alto valor'

ELSE 'Producto sin clasificar'

END AS Categoria

FROM productos;

END;

//

DELIMITER ;

CALL ClasificarProductos();

4.Este procedimiento almacenado primero obtiene el precio de venta para la venta especificada a través de la consulta.

Luego, verifica si el precio de venta es igual o superior a $100 y calcula el descuento en consecuencia. Finalmente,

actualiza el precio de venta de la venta con el descuento calculado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

DELIMITER //

CREATE PROCEDURE CalcularDescuento(IN venta\_id INT)

BEGIN

DECLARE precio\_venta DECIMAL(10, 2);

DECLARE descuento DECIMAL(10, 2);

SELECT precio\_venta INTO precio\_venta

FROM ventas

WHERE id\_venta = venta\_id;

IF precio\_venta >= 100 THEN

SET descuento = precio\_venta \* 0.10;

ELSE

SET descuento = 0;

END IF;

UPDATE ventas

SET precio\_venta = precio\_venta - descuento

WHERE id\_venta = venta\_id;

END //

DELIMITER ;

CALL CalcularDescuento(venta\_id);

Reemplaza venta\_id con el ID de la venta a la que deseas aplicar el descuento.

El procedimiento se encargará de calcular y aplicar el descuento automáticamente.

5.Este procedimiento toma un ID de venta como entrada y utiliza una consulta SQL para recuperar detalles de esa venta,

incluyendo el tipo de servicio, cantidad, precio de venta y una descripción del producto vendido.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

DELIMITER //

CREATE PROCEDURE DetallesVenta(IN venta\_id INT)

BEGIN

SELECT v.id\_venta, v.tipo\_servicio, v.cantidad, v.precio\_venta, dv.descripcion AS detalle\_producto

FROM ventas AS v

JOIN detalle\_ventas AS dv ON v.id\_venta = dv.venta

WHERE v.id\_venta = venta\_id;

END;

//

DELIMITER ;

CALL DetallesVenta(1);